EMPOWERING SECURITY, ENABLING TRUST

Black Vault Security

CONFIDENTIAL · SAMPLE REPORT

Web Application & API Penetration Test

Sample VAPT Engagement Report · v1.0

Prepared for: **Acme Media** (illustrative client)

Engagement window
04 May 2026 - 05 Jun 2026

Target
uat.acme-demo.com

Standard
OWASP Testing Guide v4.2

Prepared by
Black Vault Security

This is an **illustrative sample** built to show our reporting format. The client (Acme Media), domains and data are fictional. Evidence screenshots have been removed. · support@blackvaultsecurity.io · blackvaultsecurity.io

DOCUMENT

Confidentiality & document control

Sample notice. This document is an illustrative example of a Black Vault Security penetration-test report. The client name (Acme Media), hostnames, credentials and findings are fictional or sanitised, and all evidence screenshots have been redacted. It is provided to demonstrate our methodology and deliverable quality - it is not a record of a real engagement.

A penetration test is an interim snapshot. The findings and recommendations reflect information collected during the assessment window and do not account for changes made afterward. Time-bound engagements prioritise the controls most susceptible to exploitation rather than an exhaustive evaluation of every control.

Engagement contacts

ORGANISATION	ROLE	CONTACT
Acme Media (client)	Product Manager	alex.carter@acme-demo.com
Acme Media (client)	Senior Software Engineer	jordan.blake@acme-demo.com
Black Vault Security	Engagement Lead	support@blackvaultsecurity.io
Black Vault Security	Lead Security Consultant	support@blackvaultsecurity.io

Contents

SECTION 1

1. Executive summary

BOTTOM LINE

From **4 May 2026 to 5 June 2026**, Acme Media engaged Black Vault Security to perform a penetration test of `uat.acme-demo.com` in accordance with the OWASP Testing Guide (v4.2) and Black Vault's testing methodology. The assessment identified **17 findings**, including **1 Critical**-severity vulnerability, which has since been successfully remediated and re-verified. The findings represent exploitable weaknesses that could affect the confidentiality, integrity and availability of the application.

SCOPE

Black Vault assessed the `uat.acme-demo.com` application, its associated backend APIs and administrative dashboards. Testing was performed from Black Vault's registered accounts and IP addresses under an agreed rules-of-engagement and testing methodology.

WHAT WE FOUND

1	0	5	9	2
CRITICAL	HIGH	MEDIUM	LOW	INFO

SEVERITY	SUMMARY	STATUS
Critical ×1	A critical vulnerability that could significantly impact confidentiality, integrity and availability. Successfully remediated and verified.	Closed
Medium ×5	Input-validation, access-control, rate-limiting and client-side weaknesses exploitable under specific conditions.	Open
Low ×9	Security misconfigurations and hardening gaps with limited impact that should be addressed to improve posture.	Open
Info ×2	Best-practice observations and defence-in-depth opportunities.	Open

ENCRYPTION POSTURE

`uat.acme-demo.com` demonstrates a generally secure posture with no critical SSL/TLS vulnerabilities identified. Legacy configurations and missing hardening controls should be addressed to further strengthen resilience (see Appendix A).

PRIORITISED RECOMMENDATIONS

IMMEDIATE

Rotate all passwords and access keys exposed by the critical finding. Ensure error pages never reveal internal system details in any environment.

WITHIN 2 WEEKS

Remediate injection issues that allow script execution through admin forms, particularly in outbound email functionality.

WITHIN 30 DAYS

Harden login (CAPTCHA after failed attempts, 2-hour sessions, password history) and fix access-control gaps that expose data after logout.

SECTION 2

2. Assessment methodology

Engagements are run by senior testers and signed off by a practice lead. Testing follows recognised methodologies - OWASP WSTG/ASVS, the OWASP API Security guidance, PTES and NIST practices - combining targeted automation with manual, threat-model-driven testing. Findings are confirmed manually to remove false positives.

2.1 DISCOVERY

Map the attack surface

Enumerate the technology stack, infrastructure footprint, public metadata (DNS, certificates, cloud assets, endpoints) and map user flows, privileged functions and integration points.

2.2 API TEST STRATEGY

Baseline then probe

Capture normal request/response patterns via supported clients, then craft tests that probe authentication, authorization, business logic and input validation, including parameter tampering and IDOR/BOLA.

2.3 CONFIGURATION

API-specific risk

Test object-level authorization, parameter tampering, rate-limit bypass and tenant-isolation boundaries against OWASP API guidance, with harnesses for fuzzing and sequence-based logic flaws.

2.4 ACCESS CONTROL & SESSIONS

Server-side enforcement

Validate authentication, MFA, password recovery and horizontal/vertical authorization. Examine token issuance, expiry, logout invalidation, cookie flags and concurrent-session handling.

2.5 CRYPTOGRAPHY

Correct, modern crypto

Inventory crypto usage (TLS, data-at-rest, signing, hashing); verify strong algorithms, key management, salts/IVs and secure password hashing (Argon2/PBKDF2).

2.6 DATA VALIDATION

Injection & input handling

Test SQL/command/LDAP/XML injection, file-upload validation, path traversal and deserialization, confirming server-side validation via mutation, fuzzing and boundary testing.

2.7 DATA PROTECTION

In transit, at rest, in logs

Verify TLS on sensitive flows, encryption at rest for databases/object stores, and that no PII, secrets or keys leak into logs, errors or stack traces.

2.8 ERROR HANDLING

No information leakage

Trigger unhandled exceptions to detect disclosure of stack traces, schemas, credentials or file paths; confirm fail-secure defaults and safe, generic user-facing messages.

SECTION 3

3. Severity & risk model

Severity ratings map to the CVSS v3 score ranges below and are used consistently throughout the report.

SEVERITY	CVSS V3	DEFINITION
Critical	9.0 - 10.0	Exploitation is straightforward and usually results in system-level compromise. Form a plan of action and patch immediately.
High	7.0 - 8.9	More difficult to exploit but could cause elevated privileges, data loss or downtime. Patch as soon as possible.
Medium	4.0 - 6.9	Exploitable under specific conditions or with extra steps such as social engineering. Patch after high-priority issues.
Low	0.1 - 3.9	Limited impact; reduces attack surface. Patch during the next maintenance window.
Info	N/A	Observations, strong controls noted, and best-practice documentation.

RISK = LIKELIHOOD × IMPACT

Likelihood

The potential for a vulnerability to be exploited, rated on attack difficulty, available tooling, attacker skill and the client environment.

Impact

The effect on operations - confidentiality, integrity and availability of systems and data - plus reputational and financial harm.

REVALIDATION STATUS DEFINITIONS

STATUS	MEANING
Open	The vulnerability has not been fixed.
Open (partial)	Fixed incompletely - some aspects remain, possibly at a lower severity.
Closed (exception)	Not fixed, but the risk has been formally accepted by management.
Closed	Verified to be fixed.

SECTION 4

4. Findings register

All 17 findings identified during the engagement. Representative write-ups follow in Section 5.

FINDING ID	TITLE	SEVERITY	STATUS
BVS-CRITICAL-001	Sensitive information disclosure via unhandled exception in admin change-list	Critical	Closed
BVS-MEDIUM-001	Open redirect via `next` parameter in sign-in flow	Medium	Open
BVS-MEDIUM-002	Denial of service via IP-based rate limiting on login	Medium	Open
BVS-MEDIUM-003	IP block bypass via X-Forwarded-For header manipulation	Medium	Open
BVS-MEDIUM-004	Stored cross-site scripting (XSS) in E2E products	Medium	Open
BVS-MEDIUM-005	CSS injection allows unauthorised UI manipulation	Medium	Open
BVS-LOW-001	Stored HTML injection in admin panel (product, channel, stream fields)	Low	Open
BVS-LOW-002	Hyperlink injection leading to open redirect	Low	Open
BVS-LOW-003	No password-reuse policy implemented	Low	Open
BVS-LOW-004	No rate limiting on forgot-password functionality	Low	Open
BVS-LOW-005	Broken access control - monitoring data accessible after logout	Low	Open
BVS-LOW-006	Broken access control - cached views accessible after logout	Low	Open
BVS-LOW-007	Missing rate limiting on DELETE requests across REST endpoints	Low	Open
BVS-LOW-008	Excessive session timeout allows long-lived authenticated sessions	Low	Open
BVS-LOW-009	Race condition in password-reset token handling	Low	Open
BVS-INFO-001	No rate limiting on password-change endpoint (brute-force oracle)	Info	Open
BVS-INFO-002	Account enumeration via distinct HTTP status codes (403 vs 404)	Info	Open

SECTION 5

5. Detailed findings

Representative write-ups showing our finding format. Each finding carries a description, affected surface, CIA impact, reproduction steps, remediation and references. Remaining findings in the register follow the same structure in the full report.

Critical Sensitive information disclosure via unhandled exception

`BVS-CRITICAL-001` . **Damage:** Critical . **Ease:** Easy . Closed

DESCRIPTION

The application exposes sensitive internal configuration through an unhandled exception in the Django admin panel. With `DEBUG=True` on a publicly reachable environment and `ALLOWED_HOSTS=['*']`, manipulating an object `id` on a stream-status endpoint triggers a full Django debug page that reveals the application's settings module, including credentials and infrastructure details.

AFFECTED ENDPOINT

```
GET /stream_status_e2e_v2/demo/706/channel_vis,key,nodes,status_pct,...,search/?Stream=&display_mode=full
```

IMPACT

CONFIDENTIALITY
Leak of PostgreSQL credentials, RDS hostname, ECS task ARN (AWS account ID), internal domains and S3 bucket names.

INTEGRITY
DB access enables data modification and potential privilege escalation via backend manipulation.

AVAILABILITY
Admin panel can be broken via malicious input; exposed infrastructure enables chained attacks.

REPRODUCTION

1. Log in to a normal user panel and switch context to "Demo End-2-End".
2. Capture the stream-status endpoint request shown above.
3. Change the object `id` (e.g. `714 → 752`).
4. An error page renders, leaking database username, password and other configuration.

[evidence screenshot redacted in this public sample - Fig 1: sensitive data exposure via error page]

REMEDIATION

- **Immediate:** set `DEBUG=False`; restrict `ALLOWED_HOSTS` to the real host; rotate all exposed credentials.
- **Long term:** block deploys when `DEBUG=True` outside dev, monitor for debug pages, and apply least-privilege database roles.

REFERENCES

CWE-209 (error message containing sensitive information) · CWE-200 (exposure of sensitive information) · CWE-798 (hard-coded credentials)

Retest: Fixed and verified - `DEBUG` set to false; error pages no longer leak information.

Medium Open redirect via `next` parameter in sign-in flow

BVS-MEDIUM-001 . **Damage:** Medium . **Ease:** Easy . Open

DESCRIPTION

The `/signin` endpoint accepts a `next` parameter that sets the post-authentication redirect. It is not validated, so after login users can be sent to arbitrary external URLs - enabling phishing, credential harvesting and abuse of the trusted application domain.

AFFECTED ENDPOINT

```
GET /signin/?next=<url>
```

IMPACT

CONFIDENTIALITY
Users redirected to look-alike attacker pages may disclose credentials or sensitive data.

INTEGRITY
Attackers manipulate navigation flow, steering users into unintended actions.

AVAILABILITY
Reputational and compliance risk from phishing conducted under a trusted domain.

REPRODUCTION

1. Navigate to `https://uat.acme-demo.com/signin/?next=https://attacker.example`.
2. Set `next` to any external URL.
3. Authenticate with valid credentials.
4. Observe redirection to the attacker-controlled domain after login.

[evidence screenshots redacted in this public sample - Fig 1-2: crafted URL & redirect]

REMEDIATION

- Allow only relative paths (e.g. `/dashboard`) and reject absolute external URLs.
- Enforce a strict allowlist of trusted redirect domains.
- Use framework helpers such as Django's `url_has_allowed_host_and_scheme`.

REFERENCES

CWE-601 (open redirect) · OWASP A01:2021 · OWASP Unvalidated Redirects & Forwards Cheat Sheet

Medium Denial of service via IP-based rate limiting on login

BVS-MEDIUM-002 . **Damage:** Medium . **Ease:** Easy . Open

DESCRIPTION

Login rate limiting blocks the originating IP after repeated failed attempts. Because it is enforced purely at the IP level, an attacker can deliberately fail logins from a shared/static organisational IP and lock out every legitimate user behind it - turning a brute-force defence into a denial-of-service primitive.

AFFECTED ENDPOINT

```
POST /signin
```

IMPACT

CONFIDENTIALITY
No direct impact; may indirectly facilitate targeted disruption.

INTEGRITY
No direct integrity impact identified.

AVAILABILITY
Legitimate users on a shared/static IP are denied access, disrupting business operations.

REPRODUCTION

1. From one IP, send repeated failed login requests with invalid credentials.
2. Reach the configured threshold; the originating IP is blocked.
3. Attempt access from the same IP - all users behind it are denied during the block window.

[evidence screenshot redacted in this public sample - Fig 1: IP-based blocking causes DoS]

REMEDIATION

- Add account-based throttling alongside IP limits so a whole network is not blocked.
- Use progressive delays and CAPTCHA after repeated failures rather than hard IP bans.
- Avoid fully blocking shared/static IPs where possible.

REFERENCES

CWE-307 (improper restriction of excessive authentication attempts) · CWE-770 (allocation without limits/throttling) · OWASP A07:2021

APPENDIX A

Transport security (TLS) posture

No critical SSL/TLS vulnerabilities were identified on `uat.acme-demo.com`. The following hardening items were noted to further strengthen transport security.

CONTROL	OBSERVATION	SEVERITY
Protocol versions	TLS 1.2 / 1.3 supported; ensure legacy TLS 1.0/1.1 are fully disabled.	Low
HSTS	Enforce `Strict-Transport-Security` with a long max-age and `includeSubDomains`.	Low
Cipher suites	Prefer modern AEAD suites (AES-GCM / ChaCha20-Poly1305); retire CBC-mode legacy ciphers.	Info
Cookie security	Set `Secure`, `HttpOnly` and `SameSite` on session cookies.	Low